

DOI:10.11918/j.issn.0367-6234.201806096

一种基于 Tent 映射的混合灰狼优化的改进算法

滕志军, 吕金玲, 郭力文, 许媛媛

(东北电力大学 电气工程学院, 吉林, 吉林 132012)

摘 要: 针对基本灰狼算法易陷入局部最优、未考虑个体自身经验等问题,本文提出一种基于 Tent 映射的混合灰狼优化算法 (grey wolf optimization algorithm based on particle swarm optimization,简称 PSO_GWO). 首先,其通过 Tent 混沌映射产生初始种群,增加种群个体的多样性;其次,采用非线性控制参数,前期递减速度慢,能够增加全局搜索能力,避免算法陷入局部最优,后期收敛因子递减速度快,增加算法局部搜索能力,从而提高整体收敛速度;最后,引入粒子群算法的思想,将个体自身经历过最优值与种群最优值相结合来更新灰狼个体的位置信息,从而保留灰狼个体自身最佳位置信息. 为验证该算法的有效性,本文借助 9 个标准测试函数来与其他三种算法进行对比. 实验结果表明,本文提出的算法比其他三种算法在单峰函数和多峰函数上搜索到的最优解更加理想;PSO_GWO 算法比 IGWO 算法(the improved grey wolf optimization algorithm)在计算时间复杂度方面效果较好;同时,随着种群规模增大,PSO_GWO 算法收敛值逐渐接近理想值. 因此,本文提出的 PSO_GWO 算法能更快搜索到全局最优解,且鲁棒性更好.

关键词: 灰狼优化算法;Tent 映射;非线性控制参数;粒子群算法;惯性权重系数

中图分类号: TP273 **文献标志码:** A **文章编号:** 0367-6234(2018)11-0040-10

An improved hybrid grey wolf optimization algorithm based on Tent mapping

TENG Zhijun, LÜ Jinling, GUO Liwen, XU Yuanyuan

(School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, Jilin, China)

Abstract: As the grey wolf algorithm is easy to fall into local optimum and lack of consideration of own experience, this paper proposes a grey wolf optimization algorithm based on particle swarm optimization (PSO_GWO). Firstly, it generates the initial population through Tent chaotic map, which increases the diversity of the population. Then, this paper adopts non-linear control parameters. Its decline speed is slow in the early stage, which can increase the global search ability and prevent the algorithm from falling into the local optimum. The decline speed is quick in the later stage, which can increase the algorithm's local search ability and improve the overall convergence speed. Finally, the idea of particle swarm optimization is introduced to update the position information of individual wolves by combining the best value of the individual with the best value of the population, so as to preserve the best position information of the wolves. In order to verify the effectiveness of the algorithm, this paper compared it with three other algorithms. The experimental results suggested that the solution searched by this paper is more ideal than the other three algorithms on the unimodal function and the multimodal function. The PSO_GWO algorithm worked better than the IGWO algorithm (the improved grey wolf optimization algorithm) in calculating the time complexity; as the population size increased, the convergence value of the PSO_GWO algorithm gradually approached the ideal value. So the proposed algorithm can quickly search the global optimal solution and has better robustness.

Keywords: grey wolf optimization algorithm; Tent chaotic map; nonlinear control parameter; particle swarm optimization; inertia weight coefficient

近几年来,随着科学技术的发展,群智能算法逐渐成为学者们研究的焦点. 由于群智能算法具有简单性,灵活性,无衍生机制和避免局部最优特性,不仅应用在计算机领域,而且也能应用在农业、冶金工程,光学工程等等. 迄今为止,研究人员对此已经提

出很多种智能算法,例如遗传算法^[1]、粒子群算法^[2-3]、差分算法^[4]等. 这些算法的共同目标是找到最佳解及更好的收敛性能. 但是,没有文献模拟以狩猎闻名的灰狼领导阶级. 因此,2014 年 Mirjalili 等^[5]提出灰狼优化算法 (grey wolf optimization algorithm,GWO),其主要是模拟灰狼社会阶级以及灰狼狩猎过程. 狼的领导层次分为 4 类: $\alpha,\beta,\delta,\omega$ 狼,其中 α,β 狼负责领导整个狼群. 狩猎过程主要由三个步骤组成:搜索,包围,攻击. 同时,Mirjalili 通过一系列标准测试函数验证 GWO 算法比 PSO 算

收稿日期: 2018-06-17

基金项目: 国家自然科学基金青年科学基金项目(61501107);
吉林省教育厅“十三五”科学研究规划项目(JJKH20180439KJ)

作者简介: 滕志军(1973—),男,教授,硕士生导师

通信作者: 滕志军,tenzhijun@163.com

法、GA 算法、DE 算法等收敛速度快, 稳定性更强。但是, GWO 算法在进行猎物攻击时易陷入停滞。Saremi 等^[6-7]提出利用动态进化种群与灰狼算法相结合, 从而提高算法的局部搜索能力。但忽略算法的全局搜索能力。Zhu 等^[8-10]通过引入差分进化算法的全局搜索来改善灰狼算法的搜索能力。为增加种群的多样性, 龙文等^[11]提出一种改进的灰狼优化算法 (the improved grey wolf optimization algorithm, IGWO), 其利用佳点集法初始化种群, 使初始种群分布均匀。同时, 郭振洲等^[12]引入动态权重策略来优化灰狼算法的性能。

但是上述算法在研究灰狼捕食过程时未考虑灰狼个体经验对整个种群的影响。因此, 本文提出一种改进的灰狼优化算法, 其主要是利用 Tent 映射初始化种群, 增加种群的多样性; 采用非线性控制参数策略, 加强权衡局部及全局搜索的能力; 同时, 引入 PSO 算法的思想改进灰狼位置更新公式, 避免种群提前寻优停滞。

1 灰狼优化算法

1.1 灰狼社会等级和捕猎行为

灰狼是处于食物链最顶端的捕食动物^[13-14], 其中大部分灰狼是群居生存, 每个种群平均 5~12 只灰狼, 并且种群中每个灰狼都有自己的角色, 因此它们具有非常严格的社会等级制度见图 1。

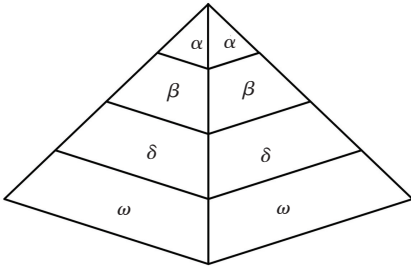


图 1 灰狼社会等级制度

Fig.1 Grey wolf social class system

第 1 层是灰狼中最高领导者, 称为 α , 其主要负责狩猎、栖息地等等各项决策; 第 2 层是灰狼中次位领导者, 称为 β , 其主要负责协助领导者管理群体; 第 3 层为 δ , 主要负责侦察、照顾年老体弱者、狩猎等等; 第 4 层是种群中最低等级的灰狼, 称为 ω , 其需要服从于所有其他阶级的灰狼, 看似 ω 狼不是一个重要的角色, 但对于平衡种群内部关系来说却是不可缺少的。

灰狼的种群等级在实现群体高效捕杀猎物的过程中发挥着至关重要的作用。首先, 灰狼群体进行

搜寻、跟踪、逐渐靠近猎物; 其次, 由 α 狼带领其余灰狼对猎物进行全方位包围; 然后, α 灰狼指挥最靠近猎物的 β, δ 狼进行攻击, 若猎物逃跑, 其它剩余灰狼从后方进行补给、围攻; 最终捕获猎物。

1.2 灰狼算法描述

灰狼算法是模拟灰狼的社会阶级制度以及其捕食行为, 然后利用灰狼在捕食过程中的搜索、包围以及捕猎等行为实现优化的目的。假设灰狼数量为 N , 搜索区域为 d 维, 其中第 i 只灰狼的位置可表示为: $X_i = (X_{i1}, X_{i2}, X_{i3}, \dots, X_{id})$, 在种群中适应度值最大的个体记为 α , 将适应度值排名第 2 及第 3 的对应个体记为 β 和 δ , 其它剩余的个体记为 ω , 猎物的位置对应于算法中的 α 狼的位置。

在搜索猎物过程中, 灰狼接近并包抄猎物行为的数学模型^[5] 为

$$D = |C \times X_p(t) - X(t)|, \quad (1)$$

$$X(t+1) = X_p(t) - A \times D. \quad (2)$$

式中: $X_p(t)$ 为第 t 代猎物的位置; $X(t)$ 为第 t 代灰狼个体的位置; C 为摆动因子, 由下式决定^[5]:

$$C = 2r_1. \quad (3)$$

式中: r_1 为 $[0, 1]$ 之间的随机数; A 为收敛因子, 由下式决定^[5]:

$$A = 2ar_2 - a, \quad (4)$$

$$a = 2(1 - \frac{t}{T_{\max}}). \quad (5)$$

式中: $T_{\max}$ 为最大迭代数目; r_2 为 $[0, 1]$ 之间的随机数; a 为控制参数, 其值随着迭代次数的增加而线性减小, $a_{\max} = 2, a_{\min} = 0$ 。

当灰狼搜索到猎物所在位置时, 首先, 头狼 α 带领狼群对猎物进行包围; 然后, 头狼 α 带领 β 狼和 δ 狼对猎物进行攻击捕捉。在灰狼群体中, α, β, δ 狼距离猎物位置最近, 因此可以通过这三者的位置来计算灰狼个体向猎物移动的位置, 其具体的数学模型如下:

$$D_\alpha = |C_1 \times X_\alpha(t) - X(t)|, \quad (6)$$

$$D_\beta = |C_2 \times X_\beta(t) - X(t)|, \quad (7)$$

$$D_\delta = |C_3 \times X_\delta(t) - X(t)|. \quad (8)$$

由式(6)~(8)计算出其余灰狼个体与灰狼 α, β, δ 的距离。

$$X_1(t) = X_\alpha(t) - A_1 \times D_\alpha, \quad (9)$$

$$X_2(t) = X_\beta(t) - A_2 \times D_\beta, \quad (10)$$

$$X_3(t) = X_\delta(t) - A_3 \times D_\delta. \quad (11)$$

由式(9)~(11)计算出其余灰狼个体移动的方向。

$$X(t+1) = \frac{X_1(t) + X_2(t) + X_3(t)}{3}. \quad (12)$$

然后由式(12)更新灰狼个体的位置。

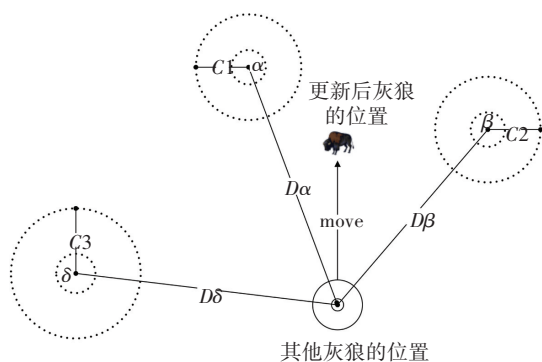


图 2 灰狼位置更新原理

Fig.2 Grey wolf location update schematic

从图 2 能够直观地看出灰狼个体根据 α, β, δ 的位置信息移动自身的位置。

1.3 算法流程

基本 GWO 算法流程见图 3。算法的主要步骤：

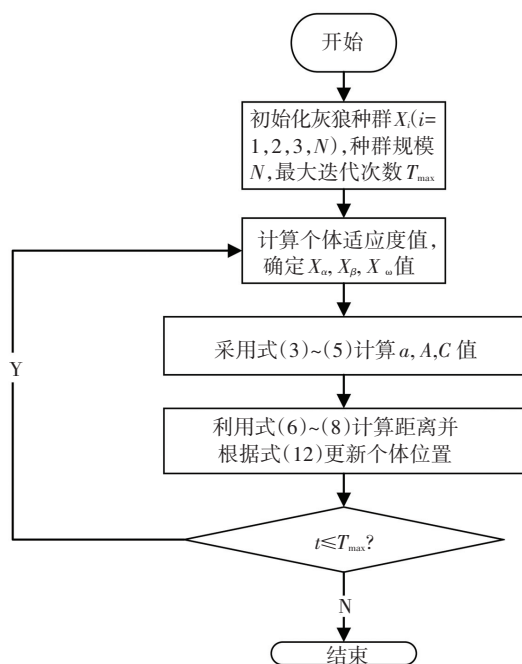


图 3 GWO 算法优化流程图

Fig.3 GWO algorithm optimization flow chart

Step1: 初始化算法参数, 即设置种群的规模为 N , 维数 d , 最大迭代次数 $T_{\max}$, 参数 A, C, a 值;

Step2: 初始化种群个体: 随机产生种群个体 $\{X_i, i = 1, 2, 3 \dots N\}$;

Step3: 根据适应度函数, 计算种群中每一个灰狼的适应度值 $\{f_i, i = 1, 2, 3 \dots, N\}$, 然后将按照适应度值大小进行顺序排列, 同时将适应度值排在前三位的灰狼个体设置为 α, β, δ , 其对应的位置信息分别为: $X_\alpha, X_\beta, X_\delta$;

Step4: 利用式(6) ~ (8) 计算其他剩余灰狼与灰狼 α, β, δ 的距离, 然后根据式(9) ~ (11) 计算其余灰狼移动的方向, 最后利用式(12) 对其余灰狼位

置进行更新。

Step5: 根据适应度函数重新计算种群中灰狼个体的适应度值, 并对适应度值进行排序, 更新灰狼 α, β, δ 及其对应的位置信息。

Step6: 判断 t 是否达到 $T_{\max}$ 值, 如果达到则输出最佳解即 α 值的适应度值, 否则返回 Step4 继续执行。

2 改进的灰狼算法

2.1 Tent 混沌映射

GWO 算法在解决函数优化问题中, 通常利用随机产生的数据作为初始种群信息, 这将难以保留种群的多样性, 会造成算法的寻优结果较差。然而, 混沌运动具有随机性、规律性、以及遍历性的特征, 在求解函数优化问题时这些特性能够使算法容易逃离局部最优解, 从而可以维持种群的多样性, 同时提高全局搜索能力。现有的混沌映射有 Tent 映射、Logistic 映射等。但是, 不同的混沌映射对于提高函数优化能力不同。目前文献中大多数采用 Logistic 映射, 但是该映射在 $[0, 0.1]$ 和 $[0.9, 1]$ 两个区域内有较高的取值率。因而, Logistic 映射遍历不均匀性会导致算法的收敛速度较慢, 从而导致算法的效率降低。单梁^[15]等证明 Tent 映射比 Logistic 映射的遍历均匀性更好以及能够提高算法的寻优速度, 与此同时能够在 $[0, 1]$ 之间产生分布较均匀的初始值。因此, 本文将利用 Tent 映射初始化种群。

Tent 映射表达式:

$$x_{t+1} = \begin{cases} x_t, & 0 \leq x_t < u; \\ \frac{1-x_t}{1-u}, & u \leq x_t \leq 1. \end{cases} \quad (13)$$

当 $u = 1/2$ 时, Tent 映射具有最典型的形式, 此时所得的序列具有均匀的分布, 对不同的参数有近似一致的分布密度。

因而, 本文引用的 Tent 混沌映射的公式为

$$x_{t+1} = \begin{cases} 2x_t, & 0 \leq x_t \leq \frac{1}{2}; \\ 2(1-x_t), & \frac{1}{2} \leq x_t \leq 1. \end{cases} \quad (14)$$

即 $x_{t+1} = (2x_t) \bmod 1$

Tent 混沌映射产生序列值的具体步骤如下所示:

Step1: 随机产生初始值 x_0 (在 $[0, 1]$ 之间且避免 x_0 落在 $(0.2, 0.4, 0.6, 0.8)$), 记入标记组, 即 $y(1) = x_0, i = 1, j = 1$;

Step2: 按式(14) 迭代产生一个 x 序列, $i = i + 1$;

Step3: If 达到最大迭代次数, 则转向 Step4; Else
if $x_i = \{0, 0.25, 0.5, 0.75\}$ 或者 $x_i = x_{i-k}, k = \{0, 1, 2, 3, 4\}$, 则按式 $x(i) = y(j+1) = y(j) + c$ 改变迭代初值, c 是随机数, $j = j + 1$; Else 返回 Step2;
Step4: 运行停止, 保留 x 序列。

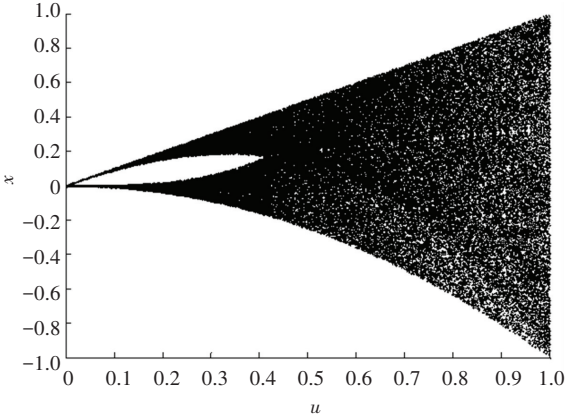


图 4 Tent 混沌映射分岔

Fig.4 Tent chaotic map bifurcation

2.2 非线性控制参数策略

GWO 算法主要通过对猎物定位以及灰狼个体移动行为两个步骤构成。根据式(2), 参数 A 对协调 GWO 算法的全局和局部开发能力具有非常关键的作用。当 $|A| > 1$ 时, 群体将扩大搜索范围, 以找到更好的候选解, 即为 GWO 算法的全局探索能力; 当 $|A| < 1$ 时, 群体将缩小搜索范围, 在局部区域进行精细搜索, 即为 GWO 算法的局部开发能力。同时, 由式(2)可知, 在进化过程中, A 值随控制参数 a 的变化而不断变化。即 GWO 算法的全局和局部开发能力取决于控制参数 a 的变化。通过上述式(4)可知, 控制参数 a 随迭代次数的增加而线性减小。然而, GWO 算法的寻优过程非常复杂, 参数 a 的线性变化不能体现算法的实际优化搜索过程。在文献[16]和文献[17]中学者们提出控制参数 a 随迭代次数非线性变化, 并通过标准测试函数优化结果表明, 采用非线性变化策略比线性策略寻优更好。文献[11]和文献[13]在此基础上也提出非线性控制参数 a , 虽然其效果有所改进, 但是仍不能满足算法的需求。

因此, 本文提出一种新的非线性控制参数为

$$a_1(t) = a_{ini} - (a_{ini} - a_{fin}) \times \left(\frac{t}{T_{max}}\right)^2. \quad (15)$$

式中: a_{ini} 和 a_{fin} 分别为控制参数的初始值及终值; t 为当前的迭代次数; T_{max} 为最大迭代数目。

为验证本文提出控制参数 a 的有效性, 将与线性控制参数以及文献[11], [13]提出非线性控制参数进行对比。线性控制参数、文献[11]和[13]的非

线性控制参数的公式为

$$a_2(t) = a_{ini} - a_{ini} \times \left(\frac{t}{T_{max}}\right), \quad (16)$$

$$a_3(t) = a_{ini} - (a_{ini} - a_{fin}) \times \tan\left(\frac{1}{\varepsilon} \times \frac{t}{T_{max}} \times \pi\right), \quad (17)$$

$$a_4(t) = a_{ini} - a_{ini} \times \left(\frac{1}{e-1} \times (e^{\frac{t}{T_{max}}} - 1)\right). \quad (18)$$

式(17)中 ε 为非线性调节系数。

本文对 4 种控制参数进行仿真见图 5, 从图 5 可看出, 本文提出的控制参数的非线性能更有效地平衡局部搜索和全局搜索能力。

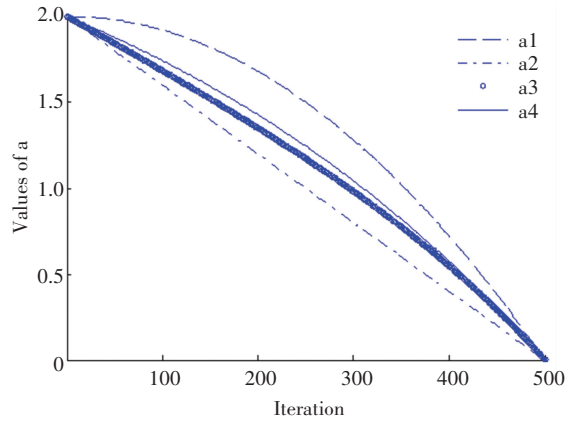


图 5 控制参数 a 的变化曲线

Fig.5 The change curve of the control parameter a

收敛因子的公式为

$$A_i = 2a_i r_2 - a_i. \quad (19)$$

式中 $i = 1, 2, 3, 4$, 分别代表不同的收敛因子。

4 种收敛因子 A 的动态变化见图 6。

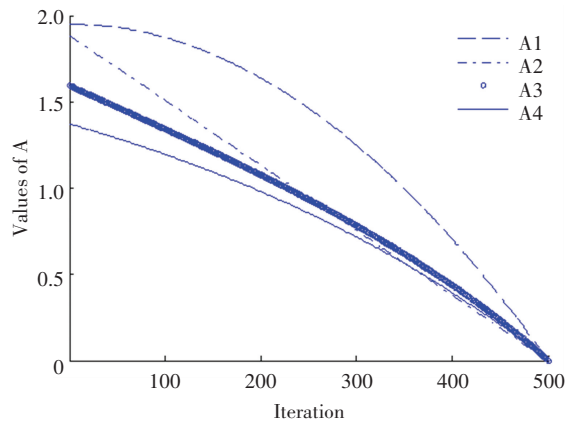


图 6 收敛因子分析动态变化曲线

Fig.6 Convergence factor dynamic curve

从图 6 可看出本文提出的收敛因子 A 前期递减速度慢, 能够增加全局搜索能力, 避免算法陷入局部最优; 后期收敛因子递减速度快, 从而能够改善局部搜索, 加快算法寻优速度。

2.3 PSO 思想

GWO 算法在位置更新过程中只考虑灰狼个体位置信息与种群的最优解、优解、次优解位置信息,并实现灰狼个体与种群之间的信息交流.但是忽略灰狼个体与自身经验之间的信息交流.因此,引入粒子群算法 (particle swarm optimization, PSO) 的思想来改进位置更新过程.

在 PSO 算法中,利用粒子自身经历过的最佳位置信息以及群体最佳位置信息来更新当前粒子的位置.因此,本文结合 PSO 算法思想将灰狼个体经历过的最优位置信息引入位置更新公式,使其能够保留自身最优位置信息.新的位置更新为

$$X_i(t+1) = c_1r_3(w_1X_1(t) + w_2X_2(t) + w_3X_3(t)) + c_2r_4(X_{ibest} - X_i(t)). \tag{20}$$

式中: c_1 为社会学习因子, c_2 为认知学习因子,其分别表示个体所经历的最优值和群体最优值对算法搜索能力的影响.如果 c_1 值太大,则会造成过多的粒子在局部附近停留;如果 c_2 值太大,则会造成粒子提前达到局部最佳值并收敛于此值.根据文献[18],本文选择 $c_1 = c_2 = 2$. r_3, r_4 为 $[0, 1]$ 之间的随机数; X_{ibest} 表示灰狼个体本身经历过最优的位置; w_1, w_2, w_3 为惯性权重系数,通过调节 α, β, δ 狼对更新个体影响的权重比例,能够动态权衡算法的全局及局部搜索能力,其具体公式如下:

$$w_1 = \frac{|X_1|}{|X_1 + X_2 + X_3|}, \tag{21}$$

$$w_2 = \frac{|X_2|}{|X_1 + X_2 + X_3|}, \tag{22}$$

$$w_3 = \frac{|X_3|}{|X_1 + X_2 + X_3|}. \tag{23}$$

2.4 PSO_GWO 算法流程

本文提出改进之后的灰狼算法 (grey wolf optimization algorithm based on particle swarm optimization, 简称 PSO_GWO) 的具体实现步骤:

Step1: 设置种群的规模为 N , 维数 d , 初始化 A, C, a 值;

Step2: 利用 Tent 映射产生种群个体 $\{X_i, i = 1, 2, 3 \dots N\}$, 同时计算各个个体的适应度值 $\{f_i, i = 1, 2, 3 \dots, N\}$;

Step3: 按照适应度值从大到小排列顺序, 取前三个适应度值所对应的个体为 α, β, δ , 其对应的位置信息分别为: $X_\alpha, X_\beta, X_\delta$;

Step4: 利用式 (15) 计算非线性变化参数 a , 然后根据式 (2)、(19) 更新 A 以及 C 值;

Step5: 利用式 (20) 更新种群个体的位置, 重新计算适应度值, 更新 α, β, δ 值;

Step6: 判断 t 是否达到 $T_{\max}$ 值, 如果达到则输出最佳解即 α 值的适应度值, 否则返回 Step3 继续执行.

表 1 PSO_GWO 算法的伪代码

Tab.1 PSO_GWO algorithm pseudo-code	
输入: 灰狼种群 X , 种群规模 N , 迭代次数 $T_{\max}$, 控制参数初值 a , 摆动因子 C , 学习因子 c_1, c_2	
输出: 最优解	
算法描述	
1: for $i = 1:N$	2: for $j = 1:d$
3: 利用式 (14) 产生 Tent 混沌序列	
4: end	5: end // 利用 Tent 初始化灰狼种群
6: for $t = 1:T_{\max}$	
7: 根据式 (15) 更新参数 a 值	8: for $i = 1:N$
9: for $j = 1:d$	
10: $D = CX_p(t) - X(t) $ 根据式 (6) (7) (8) 计算距离 D	11: 根据式 (21) (22) (23) 计算系数 w
12: 根据式 (9) (10) (11) (20) 更新灰狼个体的位置	
13: end	14: end
15: 计算适应度值并更新 α, β, δ	
16: end // 算法结束	

3 实验数据及仿真分析

3.1 基本测试函数

为验证改进之后的灰狼算法的有效性, 本文从已有文献中选择 9 个基准测试函数^[18-21]做仿真实验, 并与基本粒子群算法 (PSO)、基本灰狼算法 (GWO)^[5]、改进的灰狼算法 (the improved grey wolf optimization algorithm, IGWO)^[11] 的优化结果进行比较并分析本文算法的性能, 其具体的基准测试函数见表 2.

3.2 实验参数设置

通过 MATLAB 对上述的 9 个基准测试函数进行仿真实验对比, 具体的实验数据设置如下所示: 种群规模均为 $N = 30$; 种群维数均为 $d = 30, 50, 100$; 最大迭代数目均为 $T_{\max} = 500$; 控制参数 $a_{\min}$ 和 $a_{\max}$ 分别 2, 0; IGWO 算法中 $\varepsilon = 5, b_1 = 0.5, b_2 = 0.5$.

3.3 仿真分析

3.3.1 与基本 PSO 算法的比较

为验证本文提出的 PSO_GWO 与 PSO 算法寻优性能优劣, 对两种算法分别单独进行 30 次仿真实验, 记录其平均值以及标准差, 其结果见表 3.

表 2 基准测试函数

Tab.2 Benchmarking function

表达式	维数	搜索区间	最小值
$F_1(x) = \sum_{i=1}^{30} x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{30} x_i + \prod_{i=1}^{30} x_i $	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{30} (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq 30 \}$	30	$[-100, 100]$	0
$F_5 = \sum_{i=1}^{30} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
$F_6 = \sum_{i=1}^{30} ix_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0
$F_7 = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{30} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{30} \cos(2\pi x_i)) + 20 + e$	30	$[-5.21, 5.21]$	0
$F_8 = \sum_{i=1}^{30} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-32, 32]$	0
$F_9 = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]$	0

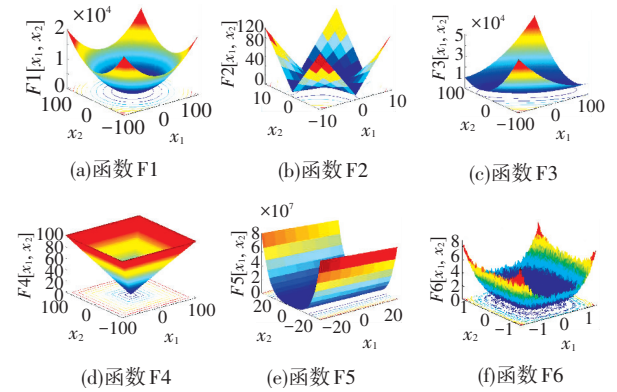


图 7 单峰基准函数

Fig.7 Unimodal benchmark functions

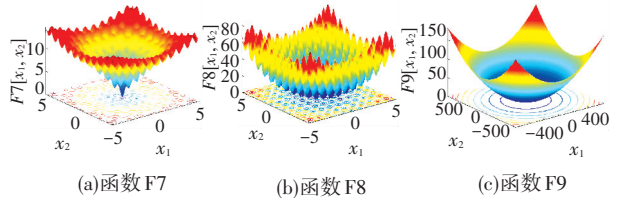


图 8 多峰基准函数

Fig.8 Multimodal benchmark functions

表 3 PSO、PSO_GWO 算法的寻优结果

Tab.3 Optimization results of PSO and PSO_GWO algorithms

函数	算法	平均值	标准差	迭代次数
F1	PSO	3.57×10^{-3}	1.02×10^{-3}	97
	PSO_GWO	4.63×10^{-59}	3.61×10^{-59}	137
F2	PSO	4.59×10^{-2}	4.21×10^2	76
	PSO_GWO	1.44×10^{-39}	5.18×10^{-39}	99
F3	PSO	38.27	2.87×10^1	87
	PSO_GWO	1.22×10^{-63}	4.86×10^{-63}	117
F4	PSO	1.98	3.26×10^{-1}	101
	PSO_GWO	4.31×10^{-31}	4.51×10^{-30}	131
F5	PSO	96.6218	1.02×10^1	85
	PSO_GWO	2.68×10^1	1.01×10^{-2}	134
F6	PSO	2.10×10^{-6}	8.97×10^{-5}	96
	PSO_GWO	2.22×10^{-8}	8.49×10^{-5}	159
F7	PSO	1.76×10^{-1}	3.01×10^{-1}	63
	PSO_GWO	1.18×10^{-17}	2.12×10^{-17}	87
F8	PSO	3.426×10^1	1.32×10^1	107
	PSO_GWO	4.16×10^{-22}	1.53×10^{-22}	124
F9	PSO	1.60×10^{-12}	2.02×10^{-12}	103
	PSO_GWO	9.31×10^{-14}	3.22×10^{-14}	121

由于基本灰狼算法在寻找猎物过程中只考虑灰狼个体位置信息与种群的最优解、次优解位置信息,但是忽略灰狼个体与自身经验之间的信息交流. 所以本文根据粒子群算法的思想将灰狼个体自身经验引入位置更新公式. 通过表 3 数据可以得到,本文提出的 PSO_GWO 算法迭代次数高于 PSO 算法,同时所得到的计算结果优于 PSO 算法的计算结果,说明算法能有效地平衡局部搜索和全局搜索能力,从而促使 PSO_GWO 算法避免在搜索过程中提前收敛而停滞现象,更好地寻找全局最优解.

3.3.2 与基本 GWO 以及改进的 GWO 算法的比较

对 GWO 算法、IGWO 算法、PSO_GWO 算法分别单独进行 30 次仿真实验,记录其平均值及标准差. 具体结果见表 4.

由表 4 可知,当维数 $d = 30$ 时,PSO_GWO 比 GWO 算法、IGWO 算法在 F1 ~ F9 测试函数上得到的平均值更加理想. 因此,PSO_GWO 算法比其他三种算法寻优性能更好. PSO_GWO 算法在 9 个测试函数中的标准差较小,说明该算法的鲁棒性较好. 同时,不管维数 30、50 还是 100,PSO_GWO 算法均比其他三种算法得到更好的最优解. 但是,随着维数的增加,PSO_GWO 算法的寻优能力略有下降.

表 4 GWO、IGWO、PSO_GWO 算法的寻优结果

Tab.4 Optimization results of GWO, IGWO, and PSO_GWO algorithms

函数	维数	GWO		IGWO		PSO_GWO	
		平均值	标准差	平均值	标准差	平均值	标准差
F1	30	3.42×10^{-27}	5.93×10^{-27}	6.56×10^{-42}	2.05×10^{-41}	4.63×10^{-59}	3.61×10^{-59}
	50	8.22×10^{-27}	4.16×10^{-27}	2.16×10^{-42}	1.31×10^{-42}	2.39×10^{-58}	4.79×10^{-58}
	100	2.24×10^{-26}	1.43×10^{-27}	8.04×10^{-41}	3.53×10^{-41}	9.21×10^{-58}	4.81×10^{-58}
F2	30	2.80×10^{-17}	5.77×10^{-17}	6.27×10^{-21}	3.71×10^{-21}	1.44×10^{-39}	5.18×10^{-39}
	50	1.63×10^{-16}	5.32×10^{-16}	2.45×10^{-20}	3.55×10^{-20}	2.53×10^{-37}	6.32×10^{-37}
	100	7.06×10^{-16}	1.87×10^{-16}	0.93×10^{-20}	1.77×10^{-20}	7.67×10^{-37}	2.46×10^{-37}
F3	30	2.44×10^{-25}	1.31×10^{-24}	1.52×10^{-42}	6.81×10^{-42}	1.22×10^{-63}	4.86×10^{-63}
	50	3.45×10^{-25}	2.29×10^{-25}	9.54×10^{-42}	2.72×10^{-42}	3.49×10^{-63}	1.56×10^{-62}
	100	3.87×10^{-25}	1.44×10^{-25}	4.03×10^{-41}	1.80×10^{-42}	3.96×10^{-62}	1.66×10^{-62}
F4	30	1.59×10^{-17}	1.41×10^{-17}	3.16×10^{-17}	1.26×10^{-17}	4.31×10^{-31}	4.51×10^{-30}
	50	1.78×10^{-16}	1.16×10^{-17}	2.06×10^{-18}	1.42×10^{-17}	3.16×10^{-31}	1.41×10^{-31}
	100	2.68×10^{-16}	2.57×10^{-16}	3.41×10^{-17}	5.78×10^{-17}	2.56×10^{-30}	4.89×10^{-30}
F5	30	3.82×10^1	4.01×10^{-2}	3.76×10^1	3.14×10^{-2}	2.68×10^1	1.01×10^{-2}
	50	3.61×10^1	3.12×10^{-2}	3.17×10^1	3.47×10^{-2}	2.70×10^1	4.01×10^{-2}
	100	4.71×10^1	8.17×10^{-2}	3.96×10^1	2.16×10^{-2}	2.81×10^1	0.87×10^{-2}
F6	30	1.50×10^{-6}	1.22×10^{-4}	3.49×10^7	7.50×10^{-5}	2.22×10^{-8}	8.49×10^{-5}
	50	1.03×10^{-6}	1.75×10^{-5}	9.76×10^{-7}	8.90×10^{-6}	3.18×10^{-8}	1.95×10^{-7}
	100	0.58×10^{-5}	2.64×10^{-5}	1.04×10^{-7}	3.75×10^{-7}	2.17×10^{-7}	3.95×10^{-7}
F7	30	3.18×10^{-13}	1.95×10^{-14}	3.65×10^{-15}	1.82×10^{-15}	1.18×10^{-17}	2.12×10^{-17}
	50	4.90×10^{-13}	1.81×10^{-13}	3.13×10^{-14}	3.49×10^{-14}	7.82×10^{-17}	4.83×10^{-17}
	100	5.33×10^{-12}	$.17\times10^{-12}$	1.45×10^{-14}	8.61×10^{-14}	5.15×10^{-16}	2.91×10^{-16}
F8	30	3.12×10^{-16}	6.31×10^{-15}	2.02×10^{-17}	3.34×10^{-16}	4.16×10^{-22}	1.53×10^{-22}
	50	6.18×10^{-16}	5.68×10^{-15}	1.02×10^{-17}	1.34×10^{-16}	7.96×10^{-22}	8.33×10^{-22}
	100	3.12×10^{-15}	3.46×10^{-15}	5.36×10^{-16}	1.14×10^{-15}	1.21×10^{-22}	3.24×10^{-21}
F9	30	7.10×10^{-13}	7.08×10^{-13}	4.16×10^{-12}	9.51×10^{-12}	9.31×10^{-14}	3.22×10^{-14}
	50	7.91×10^{-13}	1.28×10^{-12}	1.02×10^{-12}	1.36×10^{-12}	7.71×10^{-14}	5.52×10^{-14}
	100	1.47×10^{-12}	1.16×10^{-12}	7.53×10^{-12}	2.15×10^{-12}	1.08×10^{-13}	7.06×10^{-13}

为比较 4 种算法的收敛性能,本文对 4 种算法进行仿真图对比,如图 9 和图 10 所示.

从图 9 可看出,4 种算法在搜索初期时搜索速度大致相同,但是随着迭代次数的增加,本文提出的 PSO_GWO 算法一直搜索最优解,而其他三种算法提前达到寻优停滞状态,造成寻优效果差. 因此 PSO_GWO 算法在单峰函数上寻优效果优于其他三种算法. 从图 10 可看出,本文提出的算法在多峰函数上搜索到的最优解更加理想,不易陷入局部最优.

从 CPU 和 TIC/TOC 两方面对算法进行计算时间统计. CPU 反映 CPU 全速工作时完成该进程所花费的时间,而 TIC/TOC 是用来计算程序运行的时

间. 计算结果见表 5. 从表 5 可看出,PSO_GWO 算法比 IGWO 算法在计算时间复杂度方面效果较好. 但是相比 PSO 算法与 GWO 算法,PSO_GWO 算法具有较大的计算复杂度,从而运行时间较长. 这是由于 PSO_GWO 算法是在基本 GWO 算法的基础之上引入 PSO 算法的思想,从而加大算法的计算复杂度. 因此,PSO_GWO 算法牺牲计算复杂度来提高寻优效果.

3.3.3 种群规模对算法的影响

为验证种群规模对本文算法的影响,将种群规模即 N 设置为 30、35、40、45、50,并分别单独进行 30 次仿真实验,其结果如表 6 所示.

表 5 GWO、PSO、IGWO、PSO_GWO 算法的运行时间

Tab.5 Run time of GWO, PSO, IGWO, and PSO_GWO algorithm

函数	PSO		GWO		IGWO		PSO_GWO	
	TIC/TOC	CPU	TIC/TOC	CPU	TIC/TOC	CPU	TIC/TOC	CPU time
F1	1.036 21	0.037 08	11.034 32	0.037 603	11.036 47	0.044 80	11.037 74	0.038 515
F2	1.090 12	0.048 36	11.089 61	0.047 23	11.094 55	0.051 73	11.092 00	0.048 79
F3	1.025 11	0.005 928	11.019 21	0.058 76	11.026 80	0.005 992	11.025 28	0.006 192 3
F4	1.018 95	0.001 466	11.014 59	0.001 464	11.019 950	0.001 483	11.019 32	0.001 473
F5	1.013 55	0.017 316	11.014 69	0.015 391	11.015 02	0.019 324	11.014 868	0.019 125
F6	1.026 85	0.036 240	11.023 53	0.035 18	11.031 50	0.040 136	11.026 99	0.039 902
F7	1.015 58	0.054 60	11.015 633	0.054 664	11.017 72	0.057 14	11.016 86	0.054 928
F8	1.017 7	0.024 48	11.015 91	0.021 616	11.018 62	0.025 152	11.018 25	0.024 86
F9	1.042 50	0.001 540	11.039 85	0.001 34	11.051 92	0.002 07	11.044 61	0.001 73

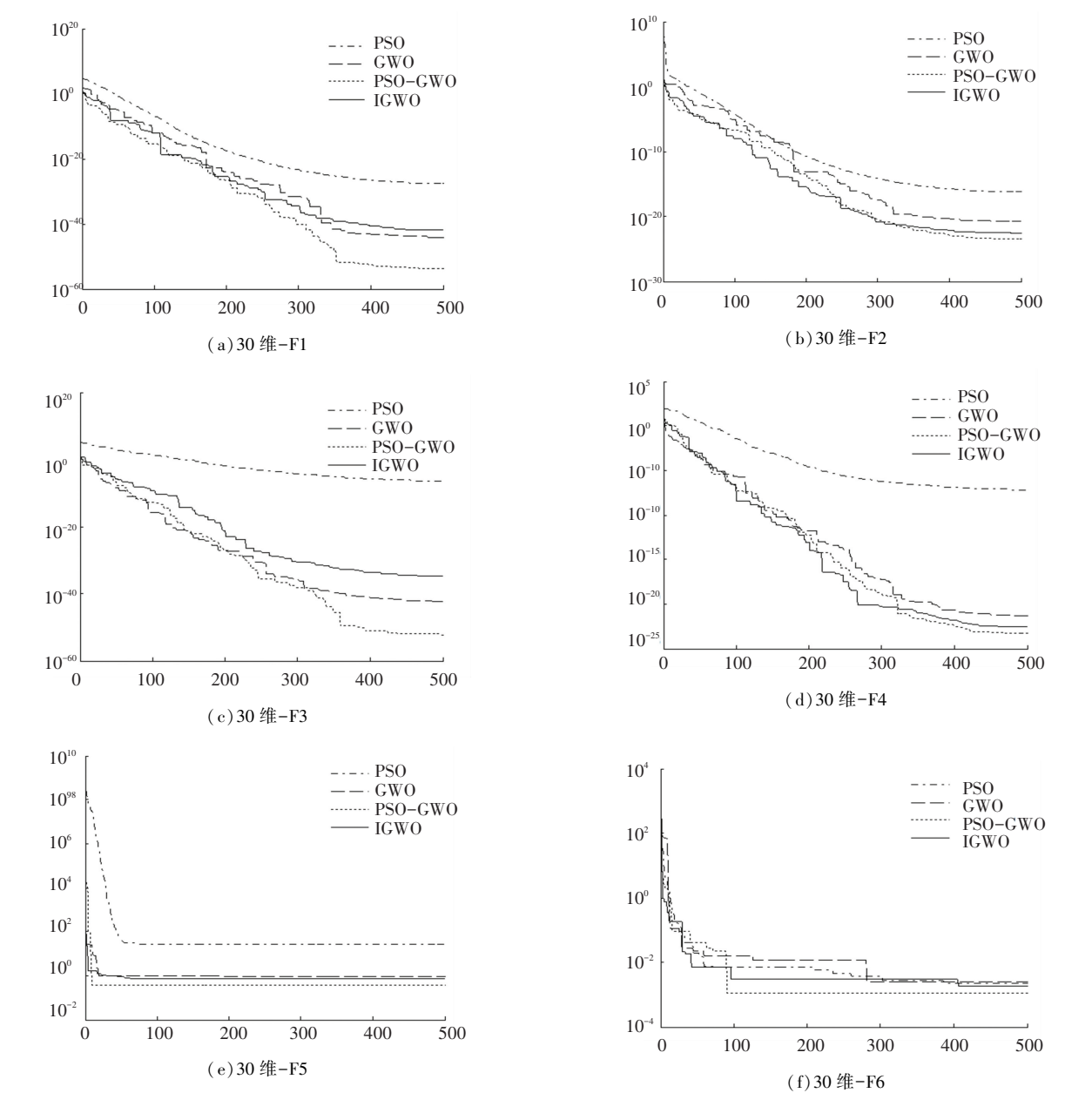


图 9 PSO、GWO、IGWO 和 PSO_GWO 在单峰函数上的收敛曲线

Fig.9 Convergence curves of PSO, GWO, IGWO, and PSO_GWO on unimodal functions

表 6 不同种群规模的 PSO_GWO 的寻优结果

Tab.6 Results of optimization of PSO_GWO for different population sizes

种群规模		PSO_GWO								
		F1	F2	F3	F4	F5	F6	F7	F8	F9
30	平均值	4.63×10^{-59}	1.44×10^{-39}	1.22×10^{-63}	4.31×10^{-31}	2.68×10^1	2.22×10^{-8}	1.18×10^{-17}	4.16×10^{-22}	9.31×10^{-14}
	标准差	3.61×10^{-59}	5.18×10^{-39}	4.86×10^{-63}	4.51×10^{-30}	1.01×10^{-2}	8.49×10^{-5}	2.12×10^{-17}	1.53×10^{-22}	3.22×10^{-14}
35	平均值	7.02×10^{-60}	1.73×10^{-40}	9.58×10^{-64}	8.25×10^{-32}	2.67×10^1	9.36×10^{-9}	0.74×10^{-15}	1.96×10^{-22}	7.87×10^{-14}
	标准差	4.95×10^{-60}	1.15×10^{-40}	6.73×10^{-63}	5.63×10^{-32}	1.72×10^{-2}	2.15×10^{-8}	3.81×10^{-16}	2.18×10^{-22}	5.71×10^{-14}
40	平均值	1.78×10^{-62}	1.39×10^{-43}	1.32×10^{-64}	1.51×10^{-35}	2.65×10^1	8.88×10^{-9}	7.95×10^{-16}	0.86×10^{-22}	1.84×10^{-14}
	标准差	9.32×10^{-61}	8.14×10^{-43}	7.96×10^{-64}	7.72×10^{-35}	1.26×10^{-2}	4.37×10^{-9}	2.06×10^{-16}	1.58×10^{-22}	6.70×10^{-14}
45	平均值	1.09×10^{-62}	4.11×10^{-44}	4.14×10^{-65}	8.63×10^{-37}	2.62×10^1	3.05×10^{-9}	3.09×10^{-16}	7.58×10^{-23}	0.11×10^{-14}
	标准差	7.68×10^{-62}	2.80×10^{-44}	2.92×10^{-65}	6.24×10^{-37}	1.46×10^{-2}	3.01×10^{-9}	6.84×10^{-17}	5.16×10^{-23}	5.27×10^{-14}
50	平均值	7.22×10^{-63}	0.14×10^{-44}	4.22×10^{-66}	1.96×10^{-37}	2.58×10^1	8.43×10^{-10}	2.38×10^{-17}	3.84×10^{-24}	8.13×10^{-15}
	标准差	3.87×10^{-63}	3.97×10^{-44}	2.16×10^{-65}	6.09×10^{-36}	1.17×10^{-2}	3.35×10^{-9}	2.39×10^{-16}	6.15×10^{-24}	3.05×10^{-15}

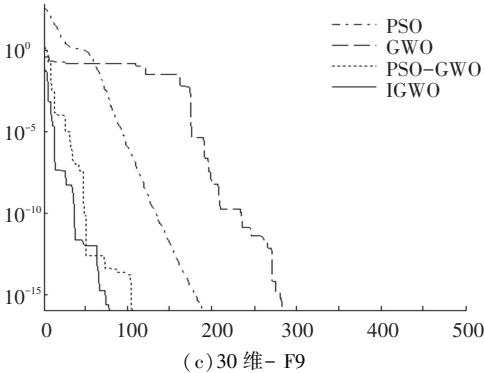
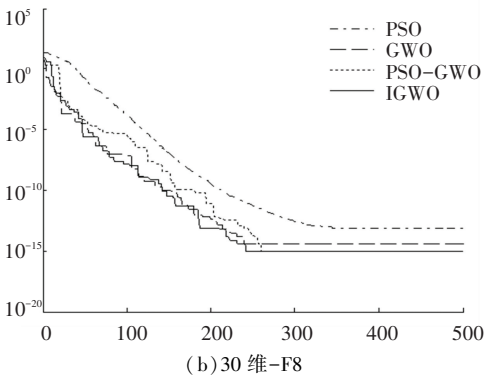
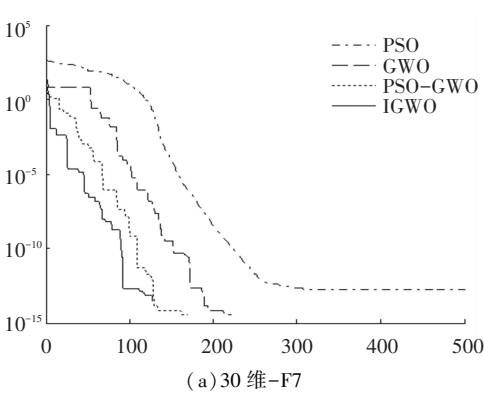


图 10 PSO,GWO,IGWO 和 PSO_GWO 在多峰函数上的收敛曲线

Fig.10 Convergence curves of PSO, GWO, IGWO, and PSO_GWO in multimodal functions

从表 6 可得到,当 $N = 50$ 时,PSO_GWO 算法的寻优结果最好;随着 N 逐渐增大,PSO_GWO 算法收敛值逐渐增大. 这是由于种群数量增加将会使算法在初始化时多样性更好,从而能够提高算法的全局能力及局部搜索能力,增强算法的寻优性能. 因此,本文提出的算法随着种群规模增大收敛值逐渐接近理想值.

4 结 论

为进一步提高算法的寻优能力及收敛精度,本文提出一种改进的灰狼算法. 其思路是利用 Tent 混沌映射初始化灰狼种群增加种群的多样性;采用非线性控制参数来权衡算法的局部及全局搜索能力,从而能够提高种群的收敛速度;与此同时,引进 PSO 算法的思想将灰狼个体经验引入位置更新公式中,使其能够保留自身最优位置信息,从而提高算法的收敛精度. 为验证本文提出的算法的有效性,对 9 个标准测试函数进行仿真实验. 实验结果显示,PSO_GWO 算法比 GWO、PSO、IGWO 算法更快搜索到全局最优解,且其鲁棒性更好. 下一阶段将围绕利用 PSO_GWO 算法来提高无线传感器网络覆盖效果开展研究.

参考文献

[1] ZAMZAMIAN S M, HOSSEINI S A, SAMADFAM M. Optimization of the marinelli beaker dimensions using genetic algorithm [J]. Journal of Environmental Radioactivity, 2017, 172: 81. DOI: 10.1016/j.jenvrad.2017.03.020.

[2] GARG H. A hybrid PSO-GA algorithm for constrained optimization problems [J]. Applied Mathematics and Computation, 2016, 274 (11): 292. DOI: 10.1016/j.amc.2015.11.001

[3] MADOLIAT R, KHANMIRZA E, POURFARD A. Application of PSO and cultural algorithms for transient analysis of natural gas pipeline [J]. Journal of Petroleum Science & Engineering, 2016, 149: 504. DOI: 10.1016/j.petrol.2016.09.042

- [4] SAKR W S, El-SEHIEMY R A, Azmy A M. Adaptive differential evolution algorithm for efficient reactive power management [J]. Applied Soft Computing, 2017, 53: 336. DOI: 10.1016/j.asoc.2017.01.004
- [5] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69 (3): 46. DOI: 10.1016/j.advengsoft.2013.12.007
- [6] SAREMI S, MIRJALILI S Z, MIRJALILI S M. Evolutionary population dynamics and grey wolf optimizer [J]. Neural Computing and Applications, 2015, 26 (5): 1257. DOI: 10.1007/s00521-014-1806-7
- [7] MIRJALILI S, SAREMI S, MIRJALILI S M, et al. Multi-objective grey wolf optimizer: A novel algorithm for multicriterion optimization [J]. Expert Systems with Applications, 2016, 47: 106. DOI: 10.1016/j.eswa.2015.10.039
- [8] ZHU Aijun, XU Chuanpei, LI Zhi, et al. Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC [J]. Journal of Systems Engineering and Electronics, 2015, 26 (2): 317. DOI: 10.1109/JSEE.2015.00037
- [9] 龙文, 蔡绍洪, 焦建军, 等. 求解高维优化问题的混合灰狼优化算法 [J]. 控制与决策, 2016, 31 (11): 1991. DOI: 10.13195/j.kzyjc.2015.1183
LONG Wen, CAI Shaohong, JIAO Jianjun, et al. Hybrid grey wolf optimization algorithm for high-dimensional optimization [J]. Control and Decision, 2016, 31 (11): 1991. DOI: 10.13195/j.kzyjc.2015.1183
- [10] 姚鹏, 王宏伦. 基于改进流体扰动算法与灰狼优化的无人机三维航路规划 [J]. 控制与决策, 2016, 31 (04): 701. DOI: 10.13195/j.kzyjc.2015.0191.
YAO Peng, WANG Honglun. Three dimensional path planning for UAV based on improved interfered fluid dynamical system and grey wolf optimizer [J]. Control and Decision, 2016, 31 (04): 701. DOI: 10.13195/j.kzyjc.2015.0191
- [11] 龙文, 伍铁斌. 协调探索和开发能力的改进灰狼优化算法 [J]. 控制与决策, 2017, 32 (10): 1. DOI: 10.13195/j.kzyjc.2016.1545
LONG Wen, WU Tiebin. Improved grey wolf optimization algorithm coordinating the ability of exploration and exploitation [J]. Control and Decision, 2017, 32 (10): 1. DOI: 10.13195/j.kzyjc.2016.1545
- [12] 郭振洲, 刘然, 拱长青, 等. 基于灰狼算法的改进研究 [J]. 计算机应用研究, 2017, 34 (12): 3603. DOI: 10.3969/j.issn.1001-3695.2017.12.019
GUO Zhenzhou, LIU Ran, GONG Changqing, et al. Study on improvement of grey wolf algorithm [J]. Application Research of Computers, 2017, 34 (12): 3603. DOI: 10.3969/j.issn.1001-3695.2017.12.019
- [13] 左剑, 张程稳, 肖逸, 等. 基于灰狼优化算法的多机电力系统稳定器参数最优设计 [J]. 电网技术, 2017, 41 (09): 2987. DOI: 10.13335/j.1000-3673.pst.2017.0011
ZUO Jian, ZHANG Chengwen, XIAO Yi, et al. Multi-machine PSS parameter optimal tuning based on Grey Wolf Optimizer algorithm [J]. Power System Technology, 2017, 41 (09): 2987. DOI: 10.13335/j.1000-3673.pst.2017.0011
- [14] SAHOO A, CHANDRA S. Multi-objective Grey Wolf Optimizer for improved cervix lesion classification [J]. Applied Soft Computing, 2016, 52: 64. DOI: 10.1016/j.asoc.2016.12.022
- [15] 单梁, 强浩, 李军, 等. 基于 Tent 映射的混沌优化算法 [J]. 控制与决策, 2005, 20 (2): 179. DOI: 10.3321/j.issn:1001-0920.2005.02.013
SHAN Liang, QIANG Hao, LI Jun, et al. Chaotic optimization algorithm based on Tent map [J]. Control and Decision, 2005, 20 (2): 179. DOI: 10.3321/j.issn:1001-0920.2005.02.013.
- [16] 魏政磊, 赵辉, 李牧东, 等. 控制参数值非线性调整策略的灰狼优化算法 [J]. 空军工程大学学报: 自然科学版, 2016, 17 (3): 68. DOI: 10.3969/j.issn.1009-3516.2016.03.013
WEI Zhenglei, ZHAO Hui, LI Mudong, et al. A grey wolf optimization algorithm based on non-linear adjustment strategy of control parameter [J]. Journal of Air Force Engineering University (Natural Science Edition), 2016, 17 (3): 68. DOI: 10.3969/j.issn.1009-3516.2016.03.013.
- [17] MITTAL N, SINGH U, SOHI B S. Modified grey wolf optimizer for global engineering optimization [J]. Applied Computational Intelligence and Soft Computing, 2016, 2016 (4598): 1. DOI: 10.1155/2016/7950348
- [18] CLERC M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization [C]// Proceedings of the 1999 Congress on Evolutionary Computation. Washington, IEEE, 2002: 1951. DOI: 10.1109/CEC.1999.785513
- [19] KAO Y T, ZAHARA E. A hybrid genetic algorithm and particle swarm optimization for multimodal functions [J]. Applied Soft Computing, 2008, 8 (2): 849. DOI: 10.1016/j.asoc.2007.07.002
- [20] LIU Tianhua, YIN Shoulin. An improved particle swarm optimization algorithm used for BP neural network and multimedia course-ware evaluation [J]. Multimedia Tools and Applications, 2016, 76 (9): 11961. DOI: 10.1007/s11042-016-3776-5
- [21] LU Chao, GAO Liang, LI Xinyu, et al. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry [J]. Engineering Applications of Artificial Intelligence, 2017, 57 (C): 61. DOI: 10.1016/j.engappai.2016.10.01

(编辑 苗秀芝)